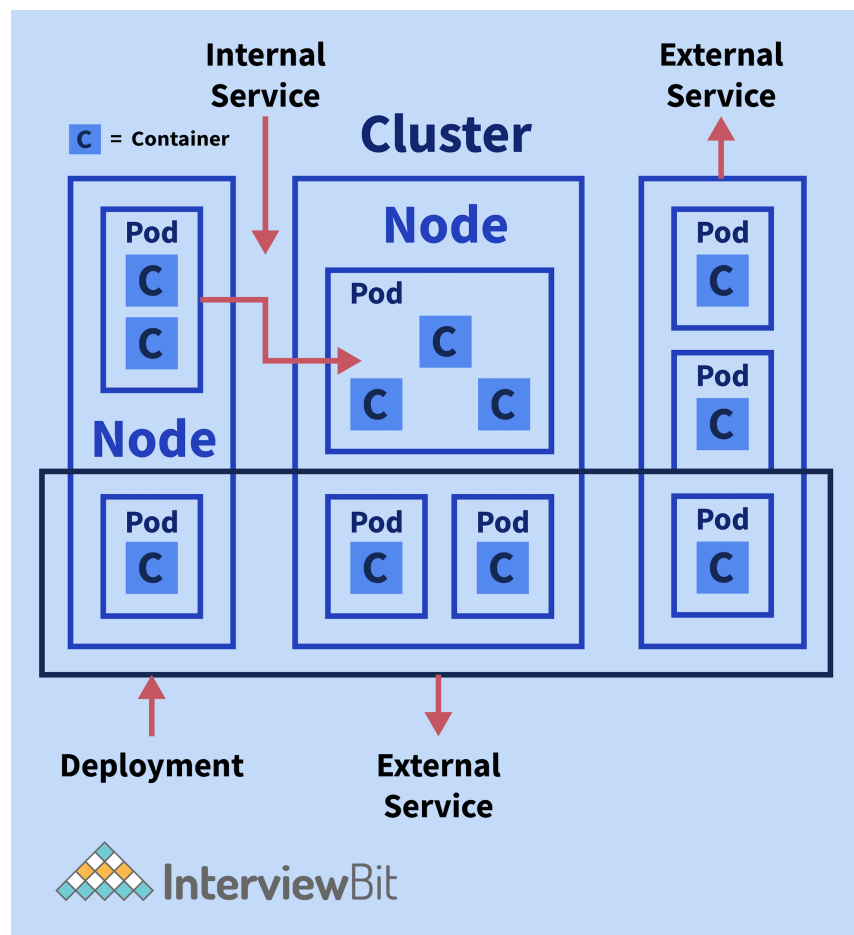


Kubernetes (Kubectl) Cheat Sheet

Credit: Interview Bit

Getting the Basics Right, Introduction to Kubernetes

Kubernetes came into the picture after the Software development teams started switching from monolithic and microservices architecture to containerization because of scalability and deployment issues. Containerization does solve the issue of scalability, downtime and dependency management quite efficiently however there are still some issues that remain unsolved.



Learn Kubernetes: Basics to Advanced Concepts

1. Kubernetes Terminology

Terms that you should be familiar with before starting off with Kubernetes are enlisted below:

Terms	Explanation
Cluster	It can be thought of as a group of physical or virtual servers where Kubernetes is installed.
Nodes	<p>There are two types of Nodes,</p> <ol style="list-style-type: none">1. Master node is a physical or virtual server that is used to control the Kubernetes cluster.2. Worker node is the physical or virtual server where workload runs in given container technology.
Pods	The group of containers that shares the same network namespaces.
Labels	These are the key-value pairs defined by the user and associated with Pods.

Master

It controls plane components to provide access points for admins to manage the cluster workloads.

Service

It can be viewed as an abstraction that serves as a proxy for a group of Pods performing a "service".

Since now we have a fair understanding of what Kubernetes is, let's now jump to the cheat sheet.

2. Kubernetes Commands

Viewing Resource Information:

1. Nodes:

ShortCode = no

A Node is a worker machine in Kubernetes and may be either a virtual or a physical machine, depending on the cluster. Each Node is managed by the control plane. A Node can have multiple pods, and the Kubernetes control plane automatically handles scheduling the pods across the Nodes in the cluster.

Commands	Description
kubectl get node	To list down all worker nodes.
kubectl delete node <node_name>	Delete the given node in cluster.
kubectl top node	Show metrics for a given node.
kubectl describe nodes grep ALLOCATED -A 5	Describe all the nodes in verbose.
kubectl get pods -o wide grep <node_name>	List all pods in the current namespace, with more details.
kubectl get no -o wide	List all the nodes with node details.
kubectl describe no	Describe the given node in verbose.
kubectl annotate node <node_name>	Add an annotation for the given node.
kubectl uncordon node <node_name>	Mark my-node as schedulable.
kubectl label node	Add a label to given node

2. Pods

Shortcode = po

Pods are the smallest deployable units of computing that you can create and manage in Kubernetes.

Commands	Description
<code>kubectl get po</code>	To list the available pods in the default namespace.
<code>kubectl describe pod <pod_name></code>	To list the detailed description of pod.
<code>kubectl delete pod <pod_name></code>	To delete a pod with the name.
<code>kubectl create pod <pod_name></code>	To create a pod with the name.
<code>Kubectl get pod -n <name_space></code>	To list all the pods in a namespace.
<code>Kubectl create pod <pod_name> -n <name_space></code>	To create a pod with the name in a namespace.

3. Namespaces

Shortcode = ns

In Kubernetes, namespaces provide a mechanism for isolating groups of resources within a single cluster. Names of resources need to be unique within a namespace, but not across namespaces.

Commands	Description
kubectl create namespace <namespace_name>	To create a namespace by the given name.
kubectl get namespace	To list the current namespace in a cluster.
kubectl describe namespace <namespace_name>	To display the detailed state of one or more namespaces.
kubectl delete namespace <namespace_name>	To delete a namespace.
kubectl edit namespace <namespace_name>	To edit and update the definition of a namespace.

4. Services

Shortcode = services

In Kubernetes, a Service is an abstraction which defines a logical set of Pods and a policy by which to access them (sometimes this pattern is called a micro-service).

Commands	Description
<code>kubectl get services</code>	To list one or more services.
<code>kubectl describe services <services_name></code>	To list the detailed display of services.
<code>kubectl delete services -o wide</code>	To delete all the services.
<code>kubectl delete service < service_name></code>	To delete a particular service.

5. Deployments

A Deployment provides declarative updates for Pods and ReplicaSets. The typical use case of deployments are to create a deployment to rollout a ReplicaSet, declare the new state of the pods and rolling back to an earlier deployment revision.

Commands	Description
<code>kubectl create deployment <deployment_name></code>	To create a new deployment.
<code>kubectl get deployment</code>	To list one or more deployments.

```
kubectl describe deployment <deployment_name>
```

To list a detailed state of one or more deployments.

```
kubectl delete deployment<deployment_name>
```

To delete a deployment.

6. DaemonSets

A DaemonSet ensures that all (or some) Nodes run a copy of a Pod. As nodes are added to the cluster, Pods are added to them. As nodes are removed from the cluster, those Pods are garbage collected. Deleting a DaemonSet will clean up the Pods it created.

Command	Description
<pre>kubectl get ds</pre>	To list out all the daemon sets.
<pre>kubectl get ds -all-namespaces</pre>	To list out the daemon sets in a namespace.
<pre>kubectl describe ds [daemonset_name][namespace_name]</pre>	To list out the detailed information for a daemon set inside a namespace.

7. Events

Kubernetes events allow us to paint a performative picture of the clusters.

Commands	Description
<code>kubectl get events</code>	To list down the recent events for all the resources in the system.
<code>kubectl get events --field-selector involvedObject.kind != Pod</code>	To list down all the events except the pod events.
<code>kubectl get events --field-selector type != Normal</code>	To filter out normal events from a list of events.

8. Logs

Logs are useful when debugging problems and monitoring cluster activity. They help to understand what is happening inside the application.

Commands	Description
<code>kubectl logs <pod_name></code>	To display the logs for a Pod with the given name.
<code>kubectl logs --since=1h <pod_name></code>	To display the logs of last 1 hour for the pod with the given name.

```
kubectl logs --tail-20 <pod_name>
```

To display the most recent 20 lines of logs.

```
kubectl logs -c <container_name>  
<pod_name>
```

To display the logs for a container in a pod with the given names.

```
kubectl logs <pod_name> pod.log
```

To save the logs into a file named as pod.log.

9. ReplicaSets

A ReplicaSet's purpose is to maintain a stable set of replica Pods running at any given time. As such, it is often used to guarantee the availability of a specified number of identical Pods.

Commands	Description
<pre>kubectl get replicaset</pre>	To List down the ReplicaSets.
<pre>kubectl describe replicaset <replicaset_name></pre>	To list down the detailed state of one or more ReplicaSets.
<pre>kubectl scale --replace=[x]</pre>	To scale a replica set.

10. Service Accounts

A service account provides an identity for processes that run in a Pod.

Commands	Description
<code>kubectl get serviceaccounts</code>	To List Service Accounts.
<code>kubectl describe serviceaccounts</code>	To list the detailed state of one or more service accounts.
<code>kubectl replace serviceaccounts</code>	To replace a service account.
<code>kubectl delete serviceaccounts <name></code>	To delete a service account.

3. Changing Resource Attributes

Taints: They ensure that pods are not placed on inappropriate nodes.

Command	Description
<code>kubectl taint <node_name><taint_name></code>	This is used to update the taints on one or more nodes.

Labels: They are used to identify pods.

Command	Description

```
kubectl label pod <pod_name>
```

Add or update the label of a pod

4. For Cluster Introspection

Commands	Description
<pre>kubectl version</pre>	To get the information related to the version.
<pre>kubectl cluster-info</pre>	To get the information related to the cluster.
<pre>kubectl config g view</pre>	To get the configuration details.
<pre>kubectl describe node <node_name></pre>	To get the information about a node.

5. Interacting with Deployments and Services

Commands	Description
<pre>kubectl logs deploy/my-deployment</pre>	Dump Pod logs for a Deployment (single-container case).
<pre>kubectl logs deploy/my-deployment -c my-contain</pre>	dump Pod logs for a Deployment (multi-container case).

```
kubectl port-forward svc/my-service  
5000
```

To listen on local port 5000 and forward to port 5000 on Service backend.

```
kubectl port-forward  
deploy/my-deployment 5000:6000
```

To listen on local port 5000 and forward to port 6000 on a Pod created by <my-deployment>.

```
kubectl exec deploy/my-deployment  
-- ls
```

To run command in first Pod and first container in Deployment (single- or multi-container cases).

6. Copy files and directories to and from containers

Commands	Description
<pre>kubectl cp /tmp/foo_dir my-pod:/tmp/bar_dir</pre>	Copy /tmp/foo_dir local directory to /tmp/bar_dir in a remote pod in the current namespace.
<pre>kubectl cp /tmp/foo my-pod:/tmp/bar -c my-container</pre>	Copy /tmp/foo local file to /tmp/bar in a remote pod in a specific container.
<pre>kubectl cp /tmp/foo my-namespace/my-pod:/tmp/bar</pre>	Copy /tmp/foo local file to /tmp/bar in a remote pod in a specific container.

```
kubectl cp
```

```
my-namespace/my-pod:/tmp/foo /tmp/bar
```

Copy /tmp/foo from a remote pod to /tmp/bar locally.

Conclusion

Kubernetes is a portable, extensible, open-source platform for managing containerization workloads and services that facilitates both declarative configuration and automation, letting you run distributed systems resiliently with scaling and failover for your application.

Knowing Kubernetes is a must-have skill whether you are a developer, a tester or a DevOps engineer, I hope this article has helped you out. If you are preparing for your interviews as a fresher or if you are an experienced person looking to switch jobs, then InterviewBit is the right place to start. It has several tracks including Programming, System Design, Puzzles and Scripting along with company-specific preparation guides and fast track courses.